Integrating Shape into an Interactive Segmentation Framework

S. Kamalakannan*[a], B. Bryant[a], H. Sari-Sarraf[a], R.Long[b] ,S. Antani[b] and G. Thoma[b]
[a]Texas Tech University, 2500 Broadway, Lubbock, TX 79401; [b]U.S. National Library of Medicine, Lister Hill National Center for Biomedical Communications, Communications Engineering Branch, 8600 Rockville Pike, Bethesda, MD 20894

ABSTRACT

This paper presents a novel interactive annotation toolbox which extends a well-known user-steered segmentation framework, namely Intelligent Scissors (IS). IS, posed as a shortest path problem, is essentially driven by lower level image based features. All the higher level knowledge about the problem domain is obtained from the user through mouse clicks. The proposed work integrates one higher level feature, namely shape up to a rigid transform, into the IS framework, thus reducing the burden on the user and the subjectivity involved in the annotation procedure, especially during instances of occlusions, broken edges, noise and spurious boundaries. The above mentioned scenarios are commonplace in medical image annotation applications and, hence, such a tool will be of immense help to the medical community. As a first step, an offline training procedure is performed in which a mean shape and the corresponding shape variance is computed by registering training shapes up to a rigid transform in a level-set framework. The user starts the interactive segmentation procedure by providing a training segment, which is a part of the target boundary. A partial shape matching scheme based on a scale-invariant curvature signature is employed in order to extract shape correspondences and subsequently predict the shape of the unsegmented target boundary. A 'zone of confidence' is generated for the predicted boundary to accommodate shape variations. The method is evaluated on segmentation of digital chest x-ray images for lung annotation which is a crucial step in developing algorithms for screening Tuberculosis.

Keywords: Interactive segmentation, Intelligent Scissors, Level sets, Tuberculosis, Lungs

1. INTRODUCTION

Currently, fully automatic segmentation techniques remain an unsolved problem due to the high level of variability and noise present in computer vision problems. In order to address this issue, certain interactive/user-driven segmentation techniques, like IS [1-2], have been developed which prove to be highly efficient for cases requiring high localization accuracy. Recently, one of the interactive segmentation techniques, namely IT-SNAPS [3-4], has been used not only for segmentation, but also to unobtrusively extract the significant features and the weight parameters necessary in order to develop an automatic segmentation procedure. However, all these interactive techniques are completely driven by low level image features and lack the advantages offered by hybrid techniques that combine low-level and high-level features. This point can be best explained by considering the examples in the next two paragraphs.

Interactive segmentation techniques like IS [1] and its generalized version Interactive Texture Snapping System (IT-SNAPS) [3-4] could be extremely burdensome for the user when the target object boundary has occlusions, broken edges or noise. Consider the Apple logo as shown in Figure 1(a) with an occluded part indicated by the green arrow. Interactive segmentation in this case will require the user to employ his/her prior knowledge about the shape of the logo in order to complete the delineation procedure. An instance of a final segmentation result and the points (yellow stars) clicked by the user are shown in Figure 1(b). This illustration highlights the necessity for employing higher level knowledge like shape in dealing with occlusions and/or broken edges to avoid increasing the burden on the user and introducing added subjectivity in the manual delineation process.

The current user-steered techniques also increase the onus on the user in the presence of strong edges that do not correspond to the actual target boundary (referred to herein as spurious edges). Consider the target object of the Texas state map outline as shown in Figure 2(a). The spurious edges in this case correspond to the internal divisions of the map outline. The user starts to segment using IS by specifying the first segment, as shown by the green boundary in Figure 2(a). Figures 2(b)-(d) show three instances of free boundary navigation (without clicking) by the user along the Texas map outline. It can be readily noticed that IS cannot distinguish between a target edge and a spurious boundary. As a result, the user is forced to place more points along the target boundary, thus increasing the burden of the segmentation

Medical Imaging 2013: Computer-Aided Diagnosis, edited by Carol L. Novak, Stephen Aylward, Proc. of SPIE
Vol. 8670, 867030 · © 2013 SPIE · CCC code: 1605-7422/13/$18 · doi: 10.1117/12.2007262

procedure. Figure 2(e) shows the final segmentation result and the points (blue stars) that the user is forced to click due to the spurious edges. The number of points the user has to click could be significantly reduced if the shape information of the Texas map can be incorporated as prior knowledge.

Such situations are commonplace in medical image segmentation applications, suggesting a need for an interactive annotation tool that can integrate shape information.

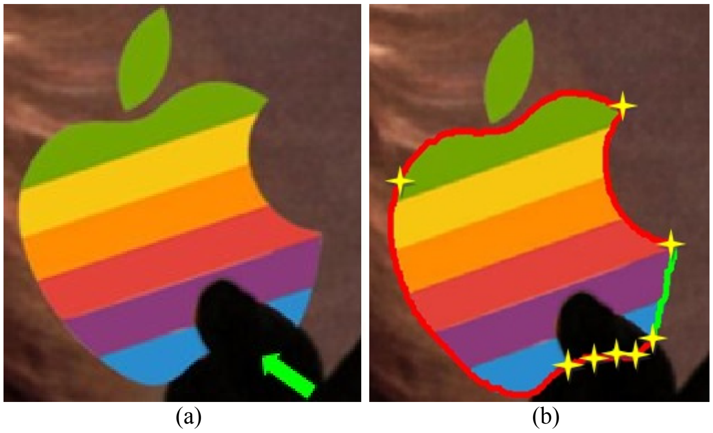
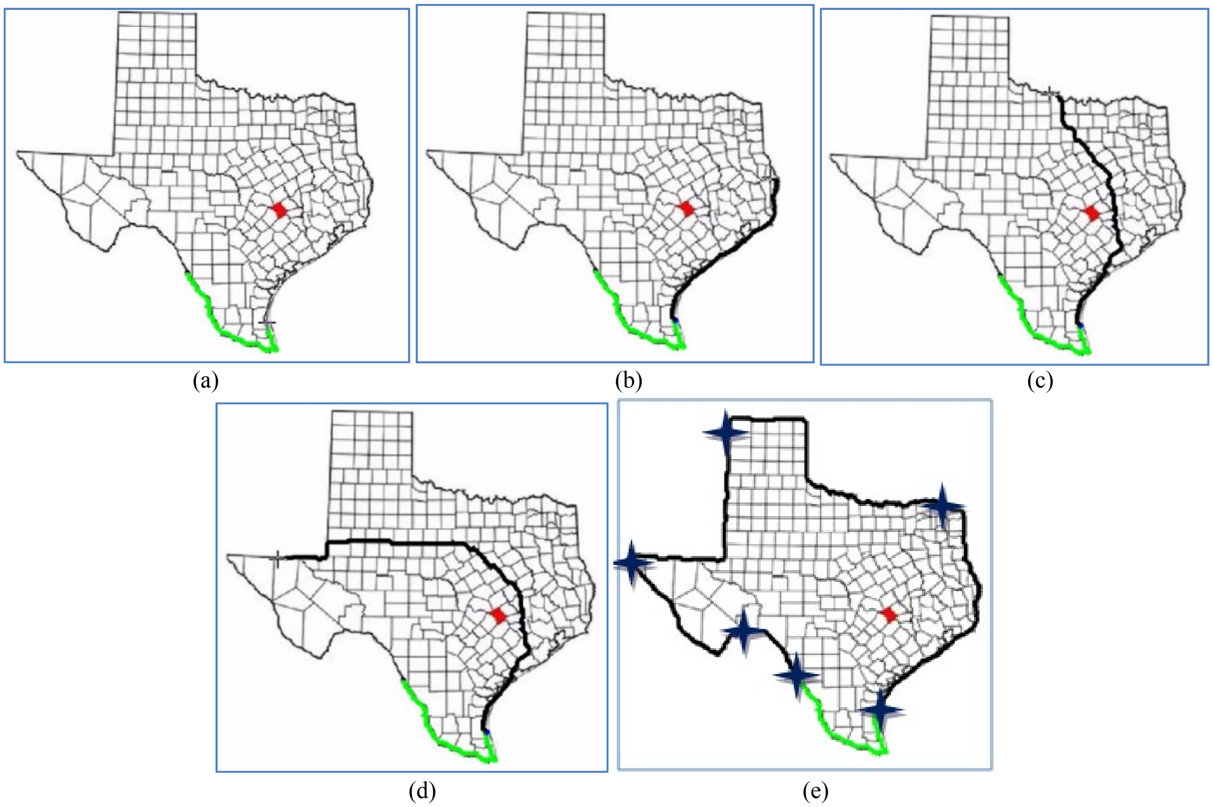

(a) (b)

Figure 1. (a) Apple logo (target boundary) with occlusion (green arrow mark); (b) Final segmentation result using IS with the user mouse clicks indicated by yellow stars


(a) (b) (c)

(d) (e)

Figure 2. (a) Texas map outline (target boundary) with spurious edges (internal divisions) with the first segment (green) specified by the user; (b)-(d) Free-hand navigation along the map outline; and (e) Final segmentation result with the user mouse clicks indicated by blue stars

This paper is organized as follows. Section 2 forms the core of the paper and discusses the proposed integration of shape into the interactive segmentation framework. Section 3 highlights the utility of the proposed work by demonstrating its use to segment occluded lung boundaries in digital chest x-ray images. Finally, Section 4 comprises some concluding remarks.

2. PROPOSED METHOD

As discussed in the previous section, embedding higher level knowledge like shape into the IS framework will mitigate the burden on the user and help the segmentation procedure. This section forms the core of this paper and will discuss all the necessary steps in order to implement shape features into IS.

As a first step, a meaningful representation of shape is extracted by computing the shape model from a set of training boundaries as discussed in Section 2.1. The user starts the interactive segmentation procedure by providing a training segment, which is a part of the target boundary. Based on this training segment, the location of the unsegmented target boundary has to be coarsely predicted. This forms the central idea of incorporating shape information into IS. This step can be thought of as superimposing the mean shape model on the target boundary by using appropriate transform parameters. In order to do this, the partial matching segment in the mean shape, corresponding to the training segment provided by the user, has to be extracted. Section 2.2 discusses the partial shape matching algorithm employed in order to extract this partial mean shape segment. Once the partial shape is extracted, the appropriate transform parameters that relate the training segment and the extracted partial shape are computed. The computed transform parameters are then applied to the full mean shape. This transformed mean shape in combination with the variance is employed to generate a 'zone of confidence' as explained in Section 2.3. The zoning procedure mitigates the influence of spurious edges and therefore reduces the burden on the user.

2.1 Shape model

The first step is to come up with a shape model for the target boundary. The shape model is generated by first aligning the training shapes employing a level set based registration framework [5]. A level-set [6] registration scheme is deemed appropriate for this application, as it does not require any point to point correspondence between the delineated shapes. A short technical description of this will help in understanding the modeling stage of the proposed approach.

The central aim is to register the level set representation, i.e. signed distance function, of a source shape $'B'$ to a target shape $'A'$. The registration is performed by seeking a global rigid transformation $'T'$, that minimizes the following energy functional,

$$E_{registration} = \sum_{(x,y)} (s\phi_B(x,y) - \phi_A(T_x, T_y))^2 \tag{2.1}$$

where ϕ_A and ϕ_B represent the signed distance functions of shapes $'A'$ and $'B'$ respectively, $'T_x'$ and $'T_y'$ represent the transformed $'x'$ and $'y'$ coordinates using the rigid transformation matrix $'T'$ respectively as follows,

$$\begin{aligned} T_x &= sx\cos(\theta) - sy\sin(\theta) + t_x \\ T_y &= sx\sin(\theta) + sy\cos(\theta) + t_y \end{aligned} \tag{2.2}$$

where, s , θ , t_x and t_y represent the scale, rotation and the translation in the vertical and horizontal directions respectively.

It can be easily noted that the registration energy term in Eq. (2.1) is minimized if the transformed source shape $'B'$ matches the target shape $'A'$, thus registering $'B'$ to $'A'$. At every iteration, the global transformation parameters are updated by minimizing Eq. (2.1) with respect to the corresponding transformation parameter and the levels-sets are evolved in parallel by minimizing Eq. (2.1) with respect to the signed distance function of the target shape $'B'$. The equations for computing the steady state solutions of the transformation parameters and the signed distance function ϕ_B are as shown below.

$$\frac{ds}{dt} = 2\sum_{(x,y)} (s\phi_B(x,y) - \phi_A(T_x,T_y))\left(\phi_B(x,y) - \left[\frac{\partial\phi_A}{\partial T_x}\frac{\partial T_x}{\partial s} + \frac{\partial\phi_A}{\partial T_y}\frac{\partial T_y}{\partial s}\right]\right) \tag{2.3}$$

$$\frac{d\theta}{dt} = -2\sum_{(x,y)} (s\phi_B(x,y) - \phi_A(T_x,T_y))\left(\frac{\partial\phi_A}{\partial T_x}\frac{\partial T_x}{\partial\theta} + \frac{\partial\phi_A}{\partial T_y}\frac{\partial T_y}{\partial\theta}\right) \tag{2.4}$$

$$\frac{dt_x}{dt} = -2\sum_{(x,y)} (s\phi_B(x,y) - \phi_A(T_x,T_y))\left(\frac{\partial \phi_A}{\partial T_x}\right) \tag{2.5}$$

$$\frac{dt_y}{dt} = -2\sum_{(x,y)} (s\phi_B(x,y) - \phi_A(T_x,T_y))\left(\frac{\partial \phi_A}{\partial T_y}\right) \tag{2.6}$$

$$\frac{d\phi_B}{dt} = 2s\sum_{(x,y)} (s\phi_B(x,y) - \phi_A(T_x,T_y)) \tag{2.7}$$

The right hand sides of Eq. (2.3) to (2.7) indicate the derivative of the right hand side of Eq. (2.1) with respect to each of the transform parameters and the evolving level set. Here $'dt'$ corresponds to the incremental time step required to iterate each parameter in order to obtain a steady state solution. The partial derivatives of $'T_x'$ and $'T_y'$ in Eq. (2.3) and (2.4) can be easily computed from Eq. (2.2). Implementing Eq. (2.3) to (2.7) in an iterative framework will register the source shape $'B'$ to the target shape $'A'$. As an illustration, the registered training shapes are overlaid on the mean signed distance shape model as shown in Figures 3 (a) and (b).

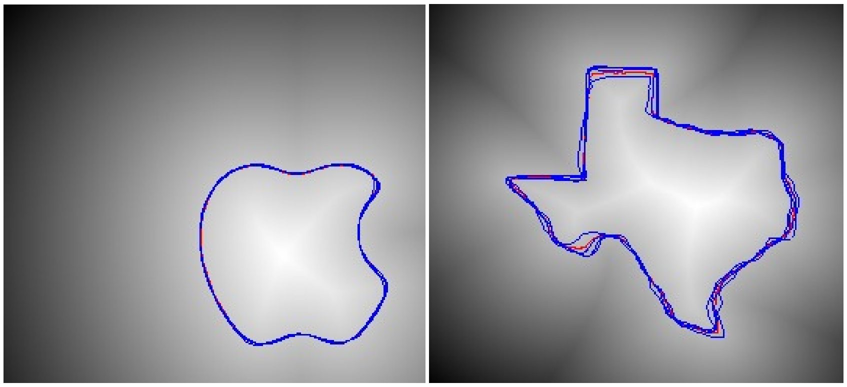


Figure 3. (a)-(b) Training shapes(in blue) displayed on the mean signed distance model after registration

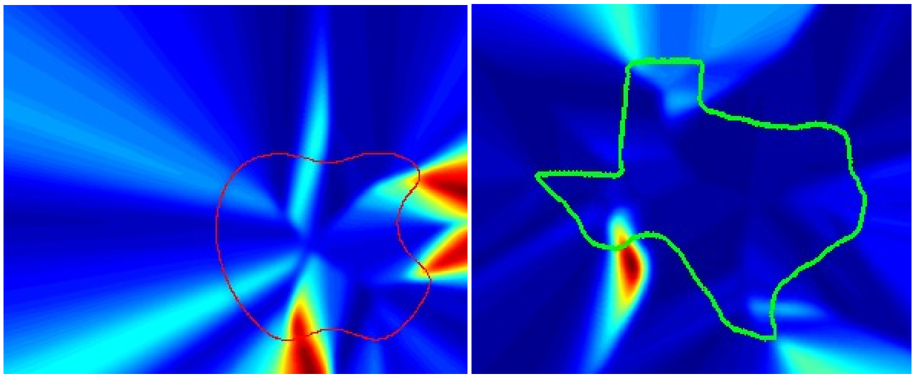


Figure 4 (a)-(b) Variance of the training shapes, with the boundary indicating the mean shape

In addition to the mean shape, variance of the shape model is also computed by employing the procedure discussed in [5]. The importance of capturing the variance of the training shapes will be discussed in Section 2.3. Let ϕ_m represent the mean distance function obtained from registering the shapes using the procedure in [5]. The variance of the training shapes σ_m^2 is computed as follows,

$$\sigma_m^2 = \frac{1}{N}\sum_{i=1}^{N} (\phi_i - \phi_m)^2 \tag{2.8}$$

where, ϕ_i represents the registered signed distance function of the i^{th} training shape.

As an illustration, the variance of these training shapes is displayed as a pseudo colored image in Figures 4 (a) and (b), with the boundary indicating the mean shape. Having extracted a meaningful representation out of the training shapes, the next two sections will discuss the incorporation of the computed mean shape and the variance into the interactive segmentation scheme.

2.2 Partial shape matching

The user starts the segmentation procedure by providing a training segment. Then the corresponding partial segment in the mean shape has to be extracted. A scale invariant signature based partial shape matching method [7] is employed in order to achieve this step.

Let the training segment and the mean shape be represented by the set of points X_t and X_m respectively. The partial mean shape segment X_p is defined as follows,

$$X_p = S(X_t) \tag{2.9}$$

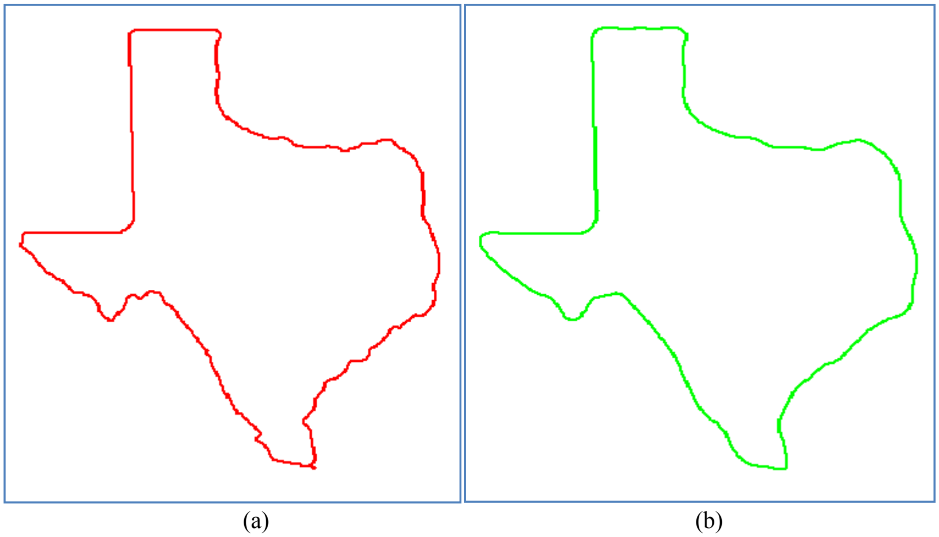
where S is any similarity transform function and $X_p \subseteq X_m$. The first step in the partial shape matching approach is to de-noise the input curves X_t and X_m. This is achieved by employing a curvature based active contour method running for a set number of iterations. The curvature snakes, whose energy term comprises only the bending energy [8], smooth the input curves X_t and X_m. An example of a mean shape and the corresponding smoothed output is displayed in Figures 5 (a) and (b), respectively. From this point onwards, X_t and X_m will refer to the smoothed versions.

After smoothing, the input curves X_t and X_m are sampled at equal arc lengths s and the corresponding curvatures, κ_t and κ_m, along these sampled points are computed as follows:

$$\begin{aligned} \kappa_t(s) &= \left\| \ddot{X}_t(s) \right\| \\ \kappa_m(s) &= \left\| \ddot{X}_m(s) \right\| \end{aligned} \tag{2.10}$$

where, ‘ $\ddot{}$ ’ correspond to the second derivative. Accordingly, $\ddot{X}_t(s)$ and $\ddot{X}_m(s)$ correspond to the second derivatives of the arc-length parameterized curves X_t and X_m. It can be shown as detailed in [7], that the integral of the absolute values of the curvatures is invariant up to a similarity transform. The integral of the curvatures from point s_1 to point s_2 is defined as,

$$K(s_1 : s_2) = \int_{s_1}^{s_2} \kappa(s) ds \tag{2.11}$$


(a)

(b)

Figure 5. (a) Mean shape and (b) Smoothed output

This integral of the absolute curvature value is the signature that is used for extracting the partial shape. The input curves X_t and X_m are now re-sampled at equal intervals of the signature K, thus making the curvatures extracted at these re-sampled points invariant up to a similarity transform. Let the new re-sampled points be represented by $X_t(K)$ and $X_m(K)$, and their corresponding curvatures at these re-sampled points be represented by $\kappa_t(K)$ and $\kappa_m(K)$, respectively. As an illustration, Figure 6 (a) shows the mean shape parameterized with respect to the arc length s and Figure 6 (b) displays the new re-sampled points of the mean shape.

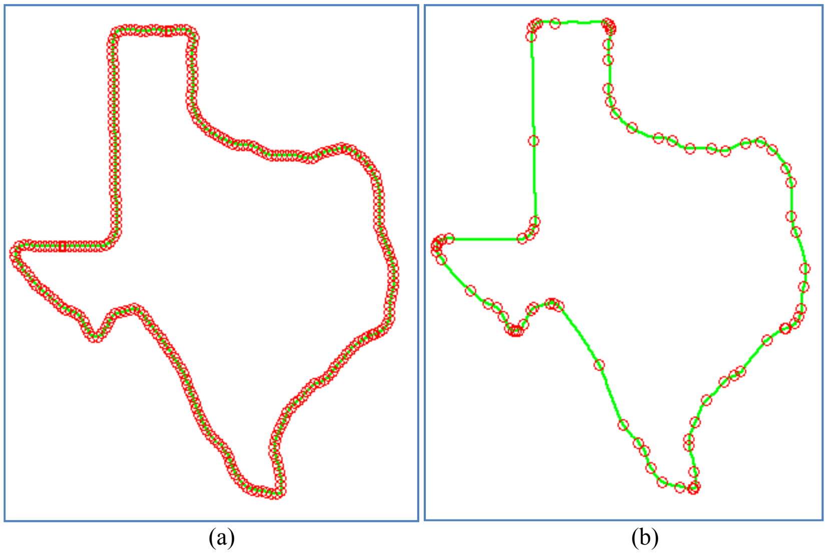

(a) (b)

Figure 6. (a) Mean shape sampled at equal arc lengths and (b) Mean shape sampled at the curvature based signature K

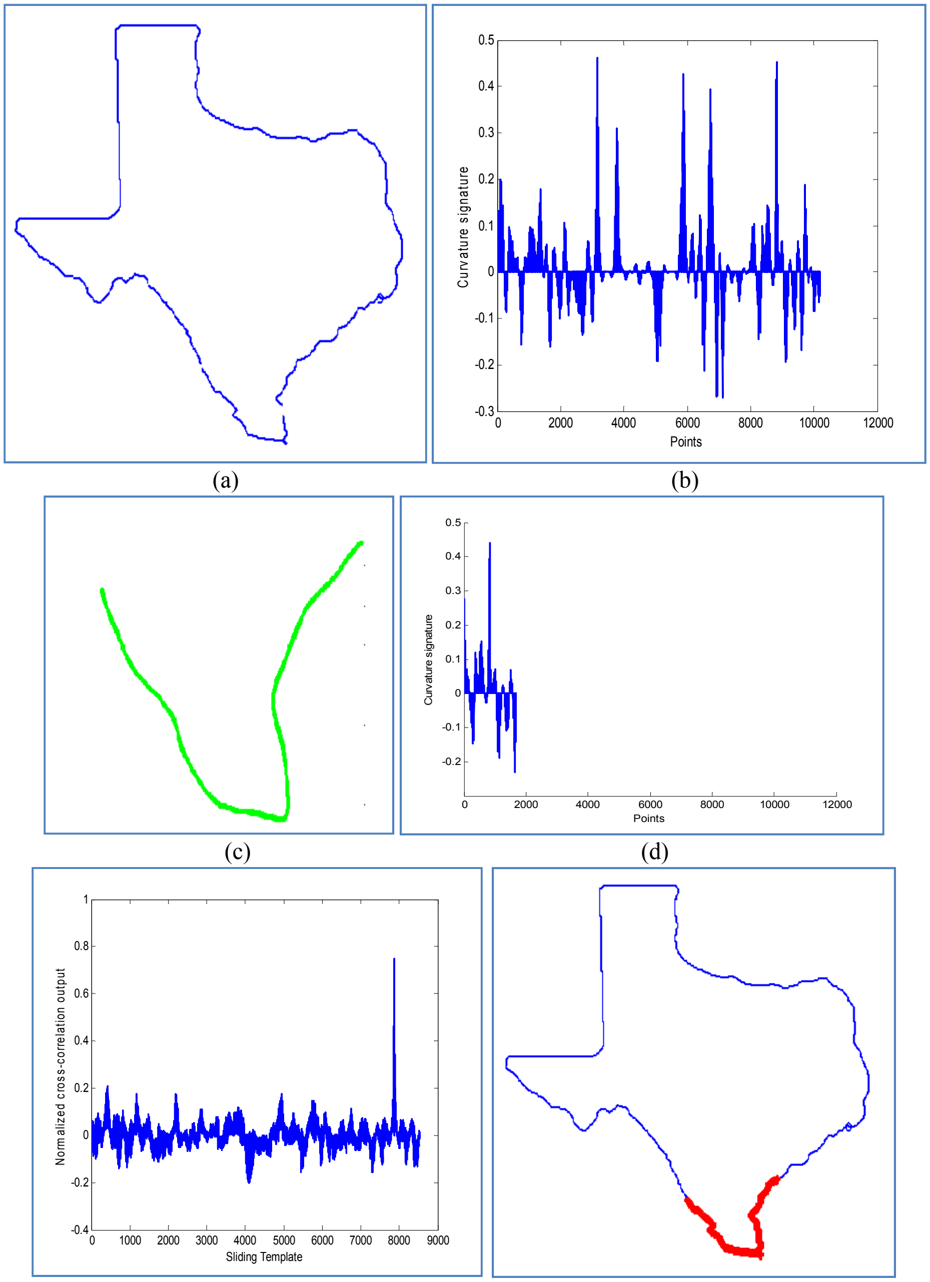
Having extracted $\kappa_t(K)$ and $\kappa_m(K)$, it now becomes a template matching problem in order to extract the partial segment X_p. Normalized cross-correlation [9] is employed to perform the template matching procedure and subsequently output the partial segment X_p based on the peak value. As an illustration, consider Figures 7 (a) and (c) which display X_m and X_t, respectively. Figures 7 (b) and (d) display the corresponding signatures $\kappa_m(K)$ and $\kappa_t(K)$. The normalized cross-correlation output plot and the partial segment corresponding to the peak are displayed in Figures 7 (e) and (f), respectively.

Once the partial shape X_p is extracted, the corresponding affine transform that relates X_p and X_t is computed. The computed affine transform is then applied on the mean shape X_m, which is utilized in order to create a 'zone of confidence' for the segmentation procedure as discussed in the next sub-section.

2.3 Zone of confidence

After transforming the mean shape, a 'zone of confidence' is built around the mean shape. The 'zone of confidence' is essentially the area of the image in which the cost computation is performed. In other words, it is a masking operation which assigns a higher cost path to all the pixels outside the zone. The zoning procedure is incorporated in order to eliminate the influence of spurious edges and therefore mitigate the burden of the segmentation procedure. As an illustration, consider Figure 8 (a), with the training segment X_t displayed in green and the transformed mean shape displayed in red. It can be readily noticed that generating a segmentation zone around this transformed mean shape would essentially eliminate most of the spurious edges inside the Texas map. Figure 8 (b) displays the result after masking the unsegmented part of the image with a fixed-width mask around the transformed mean shape.

In the above case, the unsegmented target boundary is almost perfectly captured within the confidence zone. However for shapes with high variance, a fixed width zoning might not capture all the segments of the target boundary. Hence, there is a necessity to incorporate the zoning width as a function of the shape variance in order to capture the target boundary completely. Accordingly, the width of the zone for each point in the transformed mean shape is computed as a linear function of the variance at that particular point. As an illustration Figure 9 shows the variance images with the zoning boundaries in red and the mean shape in black. In Figure 9 (a) fixed width is used for zoning and in Figure 9 (b) width as a function of the variance is used for zoning. It can be observed that the areas of higher variance have a larger zoning width, thus making this method work for even shapes with high variance.



(a) (b) (c) (d) (e) (f)

Figure 7. (a) Mean shape X_m, (b) Curvature based signature $\kappa_m(K)$ (c) Training segment X_t, (d) Curvature based signature $\kappa_t(K)$, (e) Normalized cross correlation output between $\kappa_m(K)$ and $\kappa_t(K)$ and (f) Output partial segment X_p corresponding to the peak of the normalized cross-correlation values.

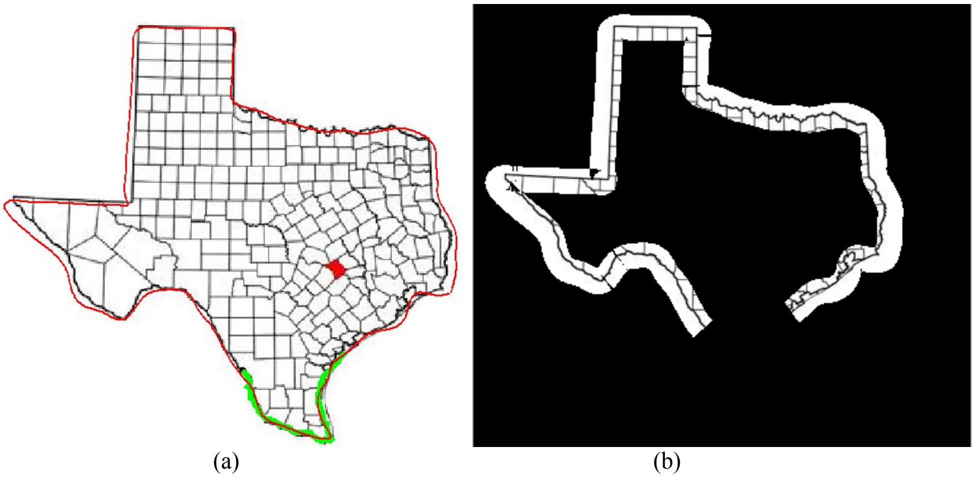


Figure 8. (a) Training segment is displayed in green and the transformed mean shape is displayed in red and (b) Result after the zoning procedure with a fixed width mask around the transformed mean shape.

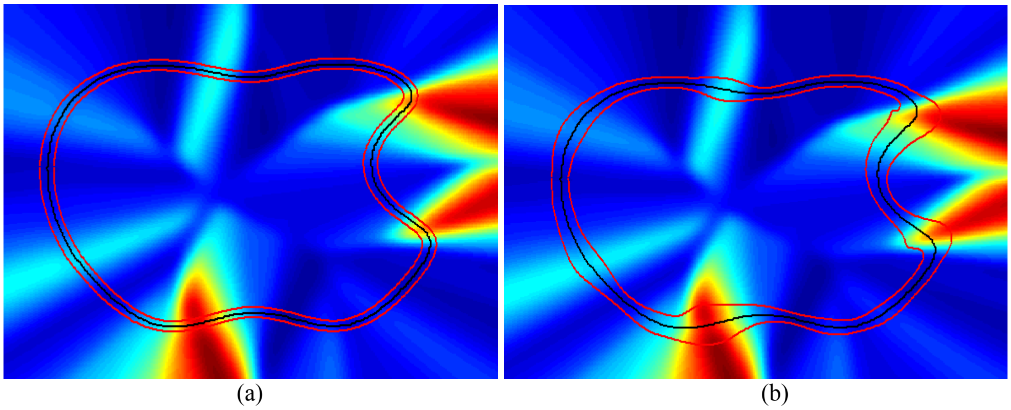


Figure 9. Variance images with the mean shape in black and the 'zone of confidence' in red (a) Fixed width for zoning and (b) Zoning width as a linear function of the shape variance

Having established the 'zone of confidence', the following section discusses the implementation of the proposed scheme into the interactive segmentation toolbox.

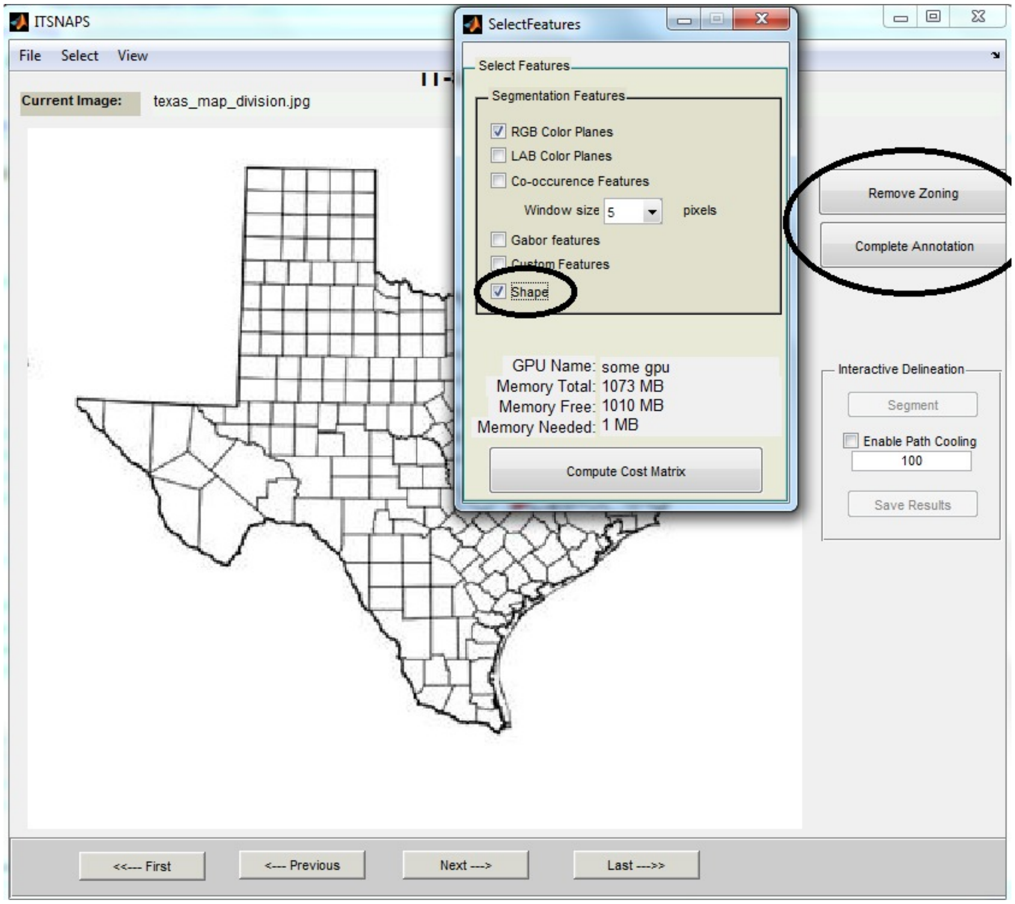
2.4 Interactive segmentation toolbox with shape feature

This section demonstrates the integration of the proposed work into the interactive segmentation toolbox. Previously, the toolbox comprised the IT-SNAPS[3-4] module, wherein the user starts the segmentation procedure by providing a training boundary based on which the appropriate features are weighted for the subsequent segmentation procedure. Currently, in addition to the image based features, the user could also select shape as a feature, provided the target boundary adheres to a shape up to a rigid transform. Once the user selects shape as a feature, he/she has to provide a pre-computed shape model or could compute and provide the shape model as discussed in Section 2.1.

The user starts the segmentation procedure by providing a training segment, which is indicated by a right click. Based on the training segment provided by the user, an appropriate partial mean shape segment and the corresponding transform parameters are determined as explained in Section 2.2. Subsequently, a zone of confidence is established as discussed in Section 2.3 which helps in mitigating the influence of spurious edges. It has to be noted that the above discussed procedure of partial shape matching and zone of confidence generation is performed at every right click the user provides. In addition, a couple of push buttons have been added to the toolbox as illustrated in Figure 10, which shows a snapshot of the toolbox.

The 'Complete annotation' button is used to fill up the occluded/broken/noisy segments of the target object. The user could segment the visible parts of the target object and press the 'Complete annotation' button in order to fill up the unsegmented parts of the target object. The utility of this function is demonstrated with real examples in Sections 2.5 and

3. The ‘Remove zoning’ button removes the zone of confidence established during the right click operation. This button is provided in case the user has provided an incorrect training segment or has clicked the right button by mistake. The next section validates the proposed toolbox on synthetic examples.



Current Image: texas_map_division.jpg

☑ RGB Color Planes
□ LAB Color Planes
□ Co-occurence Features
Window size 5 pixels
□ Gabor features
□ Custom Features
☑ Shape

GPU Name: some gpu
Memory Total: 1073 MB
Memory Free: 1010 MB
Memory Needed: 1 MB

□ Enable Path Cooling 100

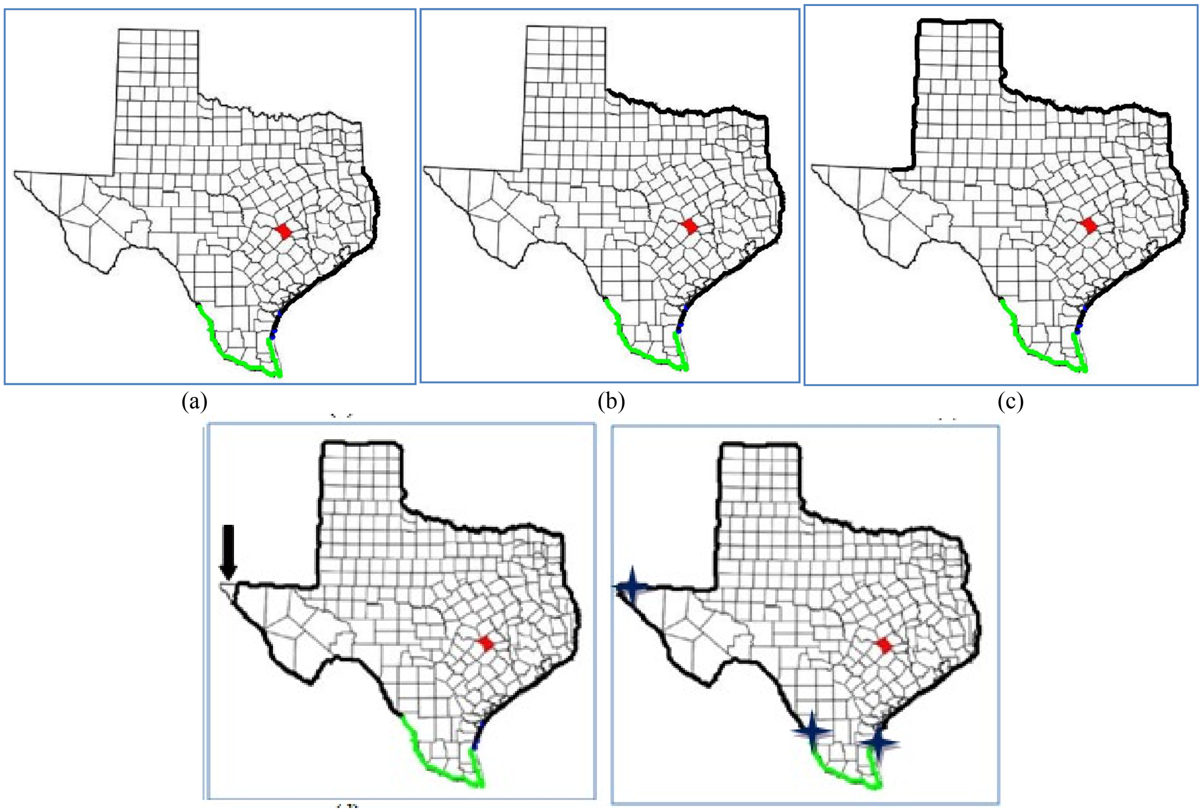
Figure 10. Snapshot of the IT-SNAPS toolbox with the new additions corresponding to the shape feature highlighted in black

2.5 Validation of the proposed approach

This section discusses the validity and the advantages of the proposed work. As discussed in Section 1, the influence of spurious edges and occlusions limit the practical application of IS. The same examples as discussed in Section 1 are used in order to illustrate the efficacy of the proposed work.

2.5.1 Spurious edges

Consider Figures 11 (a)-(d) which display the training segment in green and four instances of free boundary navigation (without clicking) by the user along the Texas map outline. It can be noticed that the influence of spurious edges is almost negligible. However, one of the spurious edges is captured within the ‘zone of confidence’ and is indicated by the arrow mark in Figure 11(d). In order to avoid this, the user places an extra point close to the spurious edge as shown in Figure 11(e). To summarize, three user clicks were required to accurately segment the map outline using shape information, whereas six user clicks were required without any shape information (See Figure 2(f)).



(a) (b) (c)

(d) (e)

Figure 11. (a)-(d) Training segment is shown in green and four instances of free boundary navigation (without clicking) by the user along the Texas map outline and (e) Final segmentation result with the user clicks displayed as blue stars

2.5.2 Occlusion/Broken edges

As illustrated in Section 1, occlusion not only increases the burden on the user, but also introduces subjectivity. In order to remove the subjectivity in the segmentation procedure, it is best if the user leaves the occluded path for the algorithm to fill up. The segmentation of the occluded region can be completed using the unsegmented part of the transformed mean shape. Consider Figure 12(a) which shows the training segment in green, and the user clicks as blue stars. The user segments the visible parts of the target boundary and leaves the occluded part for the algorithm to fill up, as displayed in Figure 12(b). The user clicks three times to segment the target, whereas eight clicks were required to segment the apple logo when no shape information is used (See Figure 2 (b)).

3. LUNG ANNOTATION

A chest x-ray screening system for pulmonary pathologies such as tuberculosis (TB) is of paramount importance due to the increasing mortality rate of patients with undiagnosed TB, especially in densely-populated developing countries. As a first step towards developing such a screening system, it is necessary to segment the lungs [10]. The segmentation task is non-trivial, especially due to occlusion of the lung region by the heart.

Figure 13 (a) shows a chest radiograph with the arrow marks indicating the lung boundaries that are occluded by the heart and Figure 13 (b) shows the lung boundaries segmented by a user using IT-SNAPS. The IT-SNAPS toolbox essentially acts as a manual segmentation tool while annotating the occluded lung segments. A test dataset of 30 chest x-ray images (60 lungs) were picked and annotated using IT-SNAPS. An average of 10 clicks were required to segment the occluded part of each of the 60 lungs using IT-SNAPS. The occluded parts for the 60 lungs were validated by a physician. The physician essentially graded the annotated occluded parts as either ‘Good’, ‘Acceptable’ or ‘Not acceptable’ boundaries. Among the 60 boundaries, 50 were graded as ‘Good’ and the remaining 10 were graded as ‘Acceptable’, thus validating the annotation procedure.

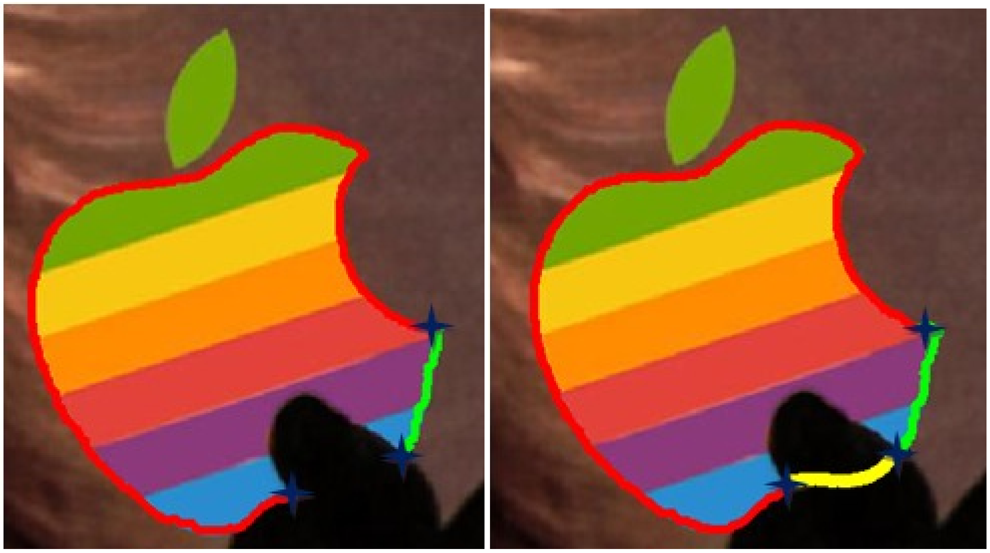


Figure 12 (a) User segments the visible target boundary and (b) The algorithm fills up the occluded part(yellow) thus completing the segmentation procedure

The proposed toolbox is employed in order to fill up the occluded boundaries without any user click. As a first step, a shape model is generated by selecting 30 chest x-ray images that are different from the previously used 30 chest x-ray images. The lung boundaries were annotated by the user for each of these 30 training images. Subsequently, a shape model was computed separately for the right and the left lungs as explained in Section 2.1. The user starts the segmentation procedure by annotating all the visible parts (in green) of the lung as shown in Figure 14.

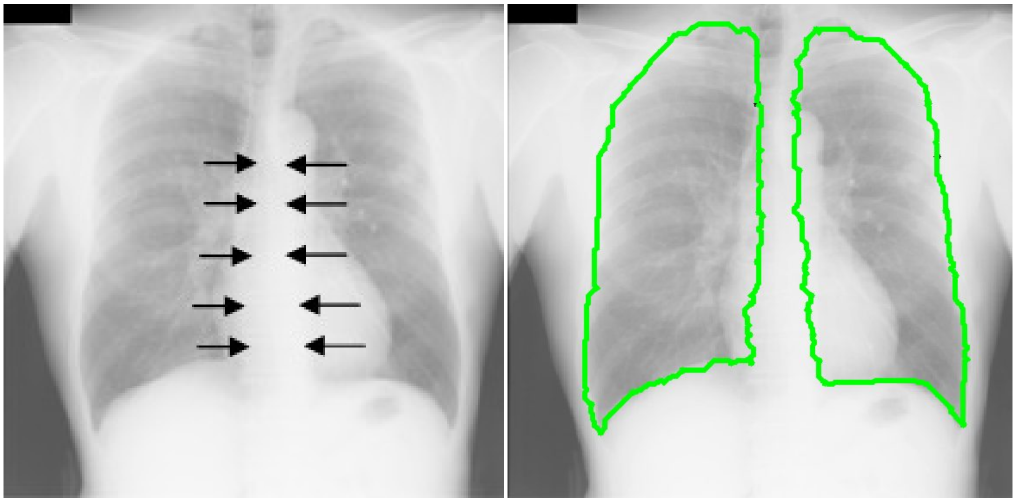


Figure 13 (a) Occluded lung boundaries by the heart are indicated with the arrow marks; (b) Segmented lung boundaries using IT-SNAPS (green)

Two experiments were conducted in order to illustrate the utility of the toolbox. Firstly, the 'Complete annotation' button, as discussed in Section 2.4, is pushed in order to fill up the occluded boundary (in pink) as shown in Figure 14. This experiment was performed on the same 30 test chest x-ray images, and the segmentation error is computed as a curve-to-curve distance between the occluded boundary segmented using IT-SNAPS and the predicted boundary by the proposed algorithm. The average error in terms of number of pixels for the 60 lungs is computed to be 5.9 pixels.

In the second experiment, the predicted segment is employed to create a zone of confidence. After the user segments the visible parts of the lung, which is treated as a training boundary, a zone of confidence is generated by employing the proposed work. The user completes the segmentation procedure by clicking on the first point, and the segmentation result of the occluded boundary (in red) is shown in Figure 15 (a). Figure 15 (b) shows an example of segmenting the occluded segment (in red) without user clicks after pushing the 'Remove zoning' button, essentially displaying the result without shape feature. It can be noticed that the boundary latches on to the heart, which in this case acts as a spurious edge. The average segmentation error by employing a zone of confidence and by asking the user to finish the segmentation procedure by clicking on the first point for the 60 lungs is computed to be 4.7 pixels. The average segmentation error obtained by pushing the 'Remove zoning' button is computed to be 12.3 pixels. An example of a segmentation error of 4.7 pixels is shown in Figure 16.

Both the above experiments reduce the burden of the segmentation procedure when compared to IT-SNAPS, thus demonstrating the utility of the proposed work.

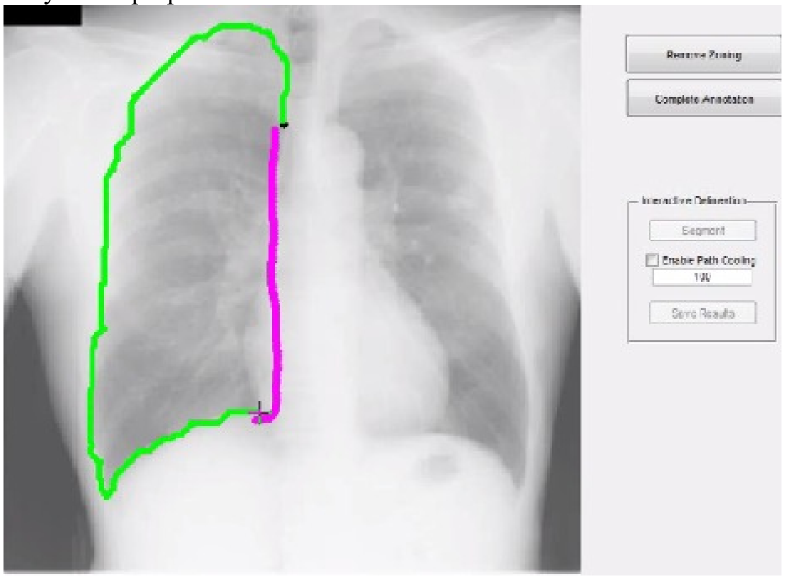
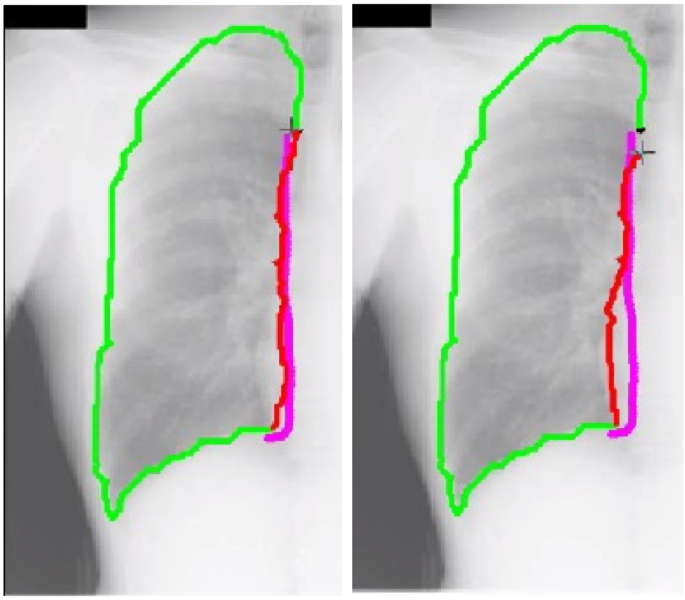


Figure 14 Algorithm fills in the occluded lung segment (pink) once the user annotates the visible boundary (green).



(a) (b)

Figure 15. Training boundary is shown in green and the predicted segments are shown in pink. Segmentation result of the occluded boundary without clicks is shown in red (a) with shape information and (b) without shape information

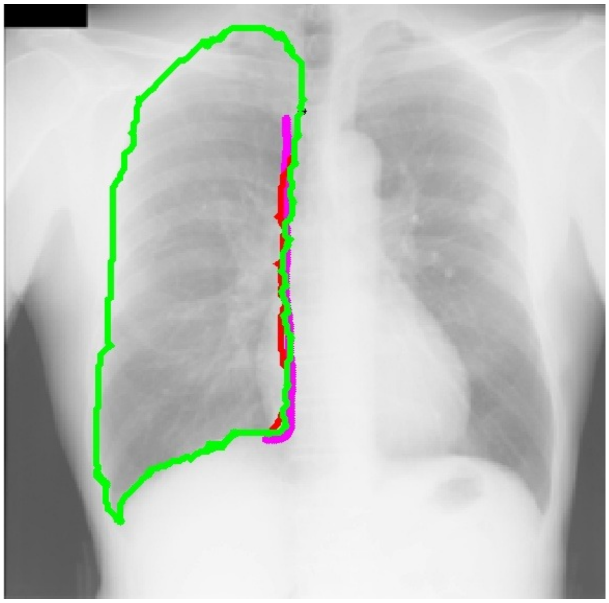


Figure 16. Segmentation error = 4.7 pixels between the occluded part of the segmented boundary (red), obtained using a zone of confidence from the predicted boundary (pink), and the occluded part of the segmented boundary using IT-SNAPS (green).

4. CONCLUSIONS

The proposed work integrates shape feature into a well known interactive segmentation framework, namely IS. Accordingly, the burden on the segmentation procedure is mitigated especially in the presence of spurious edges and occlusion. As a first step, an offline level-set based registration method is employed to determine the appropriate shape model. The user starts the segmentation procedure by providing a training boundary. A partial shape matching scheme based on a scale-invariant curvature signature is employed in order to extract shape correspondences between the training boundaries and the mean shape. Subsequently, the transform parameters that relate these matched shapes are extracted which in turn is utilized to predict the unsegmented parts of the target object; thus, reducing the burden of the annotation procedure and avoiding added subjectivity. Based on the computed transform parameters, a 'zone of confidence' is generated for the predicted boundary to accommodate shape variations. The utility of the proposed method is demonstrated on the lung annotation problem in digital chest x-ray images which is a crucial step in developing algorithms for screening tuberculosis.

ACKNOWLEDGEMENT

The authors would like to acknowledge Dr. Ragesh Panikkath, MD for validating the annotated lung boundaries. This research was supported [in part] by the Intramural Research Program of the National Institutes of Health (NIH), National Library of Medicine (NLM), and Lister Hill National Center for Biomedical Communications (LHNCBC).

REFERENCES

[1] Mortensen E. N., Barrett W. A., "Intelligent Scissors for Image Composition," Computer Graphics SIGGRAPH, 191–198 (1995)

[2] Y. Boykov, V. Kolmogorov "An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision," IEEE Trans. Pattern Anal. Mach. Intell. 26(9), 1124-1137 (2004)

[3] A. Gururajan, H. Sari-Sarraf, and E. Hequet, "Interactive Texture Segmentation Via IT-SNAPS," IEEE Proc. SSIAI IEEE Computer Society, 129-132 (2010).

[4] A. Gururajan, H. Sari-Sarraf and E. Hequet, "Generalized framework for a user-aware interactive texture segmentation system," Journal of Electronic Imaging, 21(3), (2012).

[5] M.Rousson, N.Paragios, "Prior Knowledge, Level Set Representations & Visual Grouping," Int. Journal of Computer Vision, 76(3), 231-243 (2007).
[6] S. Osher, J.A. Sethian "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," J. Comput. Phys., 79, 12–49 (1988).
[7] . M. Cui, J. Femiani, J. Hu, P. Wonka and A. Razdan, "Curve matching for open 2D curves," Pattern Recognition Letters, 30, 1-10 (2009).
[8] M. Kass, A. Witkin and D. Terzopoulos, "Active Contour Models," Int. Journal of computer vision, 1(4), 321-331 (1987).
[9] J.P.Lewis, "Fast normalized cross-correlation," In: Vision Interface. Canadian Image Processing and Pattern Recognition Society, 120–12 (1995).
[10] Karagyris A, Antani S, Thoma G, "Segmenting anatomy in chest x-rays for tuberculosis screening", EMBC, 7779-82 (2011)

.